Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Convolution Kernels for Discriminative Learning from Streaming Text

Michal Lukasik
Computer Science
The University of Sheffield
m.lukasik@shef.ac.uk

Trevor Cohn
Computing and Information Systems
The University of Melbourne
t.cohn@unimelb.edu.au

Abstract

Time series modeling is an important problem with many applications in different domains. Here we consider discriminative learning from time series, where we seek to predict an output response variable based on time series input. We develop a method based on convolution kernels to model discriminative learning over streams of text. Our method outperforms competitive baselines in three synthetic and two real datasets, rumour frequency modeling and popularity prediction tasks.

Introduction

Modeling of time series is a foundational problem in the machine learning literature, which typically attempts to model the dynamics of a signal into the future based on a history of observations. This paper considers another, less well studied, problem of discriminative modeling from time series. In this setting time series serve as inputs from which we seek to model an arbitrary output variable (or several variables). Direct modeling of time series can be viewed as a special case of such an approach, in which the response variable is the future value of the time series. However, discriminative modeling allows application to other tasks where an output variable is less closely related to the time series values. For example, we may be interested if a particular stream of posts in social media corresponds to a disaster event, or if variation of CO_2 across time in a given location indicates an alarming problem for the environment. In both instances the response variable is a classification output not directly related to the time series dynamics, however there are likely to be characteristics of the time series inputs which can be exploited in modeling the response variable.

A key problem related to discriminative modeling from time series is in capturing the similarities and differences between time series in the presence of complex temporal patterns, sporadically observed data, and non-stationarities. For instance, many collections of text data exhibit complex temporal phenomena, however these temporal dynamics are typically omitted from models. Examples include journal articles which are released at specific dates and over time their topical focus changes, or social media which exhibits trends, burstiness and a changing vocabulary.

This paper considers as its motivating case-study rumours comprising of a series of posts in conversation channels on social media. Lukasik, Cohn, and Bontcheva (2015) recently considered a discriminative learning problem of predicting rumour popularity, noting that rumour characteristics change during their lifespan: typically a rumour is initially widely discussed, but afterwards the discussion either rapidly terminates or else smoothly decays. Both the temporal dynamics of the rumour spread and the text content of the posts contain important cues for determining the future spread of a rumour. However, exploiting these two sources of information is difficult. One approach is feature engineering, however these problems are not generally amenable to feature selection, and thus result in large feature spaces with consequent high computational expense and problems of overfitting (Xing et al. 2011).

We address the problem of modeling time series of text by adapting the general idea of convolution kernels (Haussler 1999). Convolution kernels avoid the effort of feature engineering and allow for using a rich data representation. We demonstrate the proposed convolution kernels for discriminative learning from text time series on the range of problems: classification, regression, Poisson regression and point process modeling. Previous work focused on classification problems over time series (Xing, Pei, and Keogh 2010). Our approach is based on Gaussian Processes (GP) (Rasmussen and Williams 2005), which allows modelling of many different problem types, from regression modelling through to point processes.

This paper makes the following contributions: 1. Introduces a method based on convolution kernels for discriminative modeling of time series composed of text over continuous time; 2. Demonstrates the proposed approach on a range of discriminative learning problems: classification, regression, Poisson regression and point process modeling; 3. Demonstrates the efficacy of the method on three synthetic and two real datasets; and 4. Out-performs state of the art methods for modeling rumour frequency profiles.

Related Work

Discriminative Learning From Time Series Time series have been predominantly considered in settings where previous instances of a time series are used to predict the outputs for future points (Shumway and Stoffer 2006). However,

another strand of research involves discriminative learning from time series, where the aim is to perform classification over whole time series data. In general, three classes of approaches have been considered for discriminative learning from time series: model based, feature based and distance based methods (Xing, Pei, and Keogh 2010).

Model based approaches specify generative processes for the classification labels, e.g., using Hidden Markov Models (HMM) (Rabiner 1989), where a sequence of symbols is modelled as a Markov process with latent states. The HMM model has been applied for discriminative learning from biological sequences (Xing, Pei, and Keogh 2010). In this approach, an unknown sequence is being assigned a label with the highest alignment score. This method is not appropriate in our setting, because we consider time series of complex objects, namely documents, treated as bags-of-words. Moreover, we consider continuous time, rather than discrete time as in a typical HMM application.

As for the feature based methods, several approaches have been proposed. A popular approach is extraction of informative subsequence patterns (shapelets), which has been proposed in conjunction with decision trees (Ye and Keogh 2011) or SVM (Ando and Suzuki 2014). Closely related is extraction of meta features, the user-defined recurring substructures of a time series (Kadous and Sammut 2005). However, these approaches were not applied in the joint temporal and textual space. Prior work on discriminative modelling of temporal text corpora was based on feature engineering, such as descriptors of temporal pattern of text (Becker, Naaman, and Gravano 2011). Feature engineering may be troublesome and it is difficult to capture all important characteristics in the data. Firstly, the feature selection and extraction from time series data is not trivial, and secondly the feature space for time series can be high and thus computationally expensive (Xing et al. 2011). In this work we propose a kernelized approach where no explicit feature extraction is needed.

Another approach to time series classification is based on using distance functions for comparing pairs of time series. The distance function is used in conjunction with a distance based classification method such as KNN or SVM (Xing, Pei, and Keogh 2010). In case of simple time series popular approaches were euclidean distance for time series of equal lengths and Dynamic time warping (DTW) for sequences of unequal lengths (Keogh and Pazzani 2000). These kernels may be appropriate for simple time series, however, they are not adequate for this work as we deal with complex time series of text over continuous time.

Another distance based method is kernel fusion scheme (Wu, Wu, and Chang 2005), in which multiple feature extraction methods are used to produce feature representations, and modelled with several different kernels. The kernels are combined together by a weighted sum or other more complex combining function. Similarly, we consider a combination of kernels over text and over time treated as different representations of time series, which can be viewed as an instantiation of the kernel fusion scheme.[1] However, in our main contribution we do not combine kernels over different representations of time series, but use R-convolution (Haussler 1999) between kernels over individual points from them, which is a more powerful method, as we show in experimental sections.

Other examples of kernels used for modeling time series inputs include Fisher kernels (Jaakkola and Haussler 1999), graph edit distance kernels (Bunke and Allermann 1983), the probability product kernel (Jebara, Kondor, and Howard 2004) and marginalized kernels (Tsuda et al. 2001). Although designed for time series, none of the above are appropriate for our problem, since we consider multi-variate time series of text documents occurring in continuous time. None of these above algorithms were applied to multivariate time series of text. Moreover, they considered only classification settings, whereas we consider more scenarios in our experiments, such as regression, Poisson regression and point process modeling problems over time series inputs.

Text Modeling Research on text analysis focuses mainly on exploiting structure in text only, rarely making use of time. Discriminative learning over textual time series focused on symbolic sequences, where no timestamp is considered. Dynamic time warping has been applied for comparing text modelled as a time series (Liu, Zhou, and Zheng 2007; Matuschek, Schlüter, and Conrad 2008). Support vector machines with string kernels have been applied for comparing text sequences (Lodhi et al. 2002). In these approaches text is modelled as a discrete sequence, where no timestamp is assigned to them. Other relevant work includes classifying social media events (sets of posts occurring in time), and has been attempted by manual feature engineering, such as descriptors of temporal patterns of text (Becker, Naaman, and Gravano 2011).

Convolution Kernels Convolution kernels is a framework in which kernels between structured objects are specified as a combination of kernel values between substructures (Haussler 1999). Convolution kernels have been applied to discrete objects, for example graphs (Vishwanathan et al. 2010). A compelling natural language processing example is a tree kernel which operates on syntax trees through recursive decomposition into kernels between subtrees (Collins and Duffy 2001). Of particular relevance are string kernels, which recursively decompose strings into substrings (Lodhi et al. 2002). These kernels treat inputs as time series over discrete time. In this work we design a convolution kernel tailored to series of text over continuous time.

Notation and Problem Formulation

Let us consider a set of events $X = \{\mathbf{x}_i\}$, each of which consists of a set of posts $\mathbf{x}_i = \{\mathbf{p}_k^i\}$. Posts are tuples $\mathbf{p}_k^i = (\mathbf{w}_k^i, t_k^i)$, where $\mathbf{w}_k^i$ is a vector text representation and t_k^i is a timestamp describing post $\mathbf{p}_k^i$. This way, events are time series over complex objects. One can imagine different

[1] Here the different 'views' of time series are combined in a weighted average, where weights are kernel variances optimized by maximizing the evidence in the Gaussian Process framework.

situations for which the kernels we introduce can be easily adapted, e.g. using different text representation by applying string kernels over the text (Lodhi et al. 2002).

In this work we consider a range of discriminative problems, where time series are inputs and we model an output variable (most often a numerical value, but may be more complex as in case of point process below), i.e. the dataset has form $D = \{\mathbf{x}_i, y_i\}_{i=1}^n$. In particular, we consider the following four general classes of problems:

1. Binary classification, where the task is to model a binary valued function over events, $y_i \in \{0, 1\}$, e.g. labelling events as rumours or non-rumours;
2. Regression, where the task is to model a real valued function over events, $y_i \in \mathbb{R}$, e.g. price of stocks described by a time series of posts about them;
3. Poisson regression, where the task is to model a non-negative integer valued function over events, $y_i \in \mathbb{N}$, e.g. based on events up to time t, determining the number of posts in future time interval.
4. Frequency prediction, where the task is to based on the events embedded in same space, model a non-negative integer valued function over subspaces of this space; e.g. based on events up to time t, determining the number of posts in arbitrary future time intervals. In this case, y_i could be a complete set of points over the observation window.

Our kernel approach is capable of modelling a broader range of problems than those listed above. For example, it could be used in a kernelled clustering method for event detection in a social media stream. In this case, we can group together tweets based on the distance calculated using convolution kernel over text and time.

Convolution Time Series Kernels

In this section we describe the central contribution of the paper - convolution kernels between time series of posts described by text over continuous time.

Formulations

A kernel over events can be formulated in many different ways. Probably the most obvious is concatenating text from posts together and comparing resulting texts across events, e.g. using a kernel on the vectors. Such kernel can be expressed via the formula

$$K_{\Sigma\text{text}}(\mathbf{x}_i, \mathbf{x}_j) = k_{\text{text}}\left(\sum_{\mathbf{p}_k^i \in \mathbf{x}_i} \mathbf{w}_k^i, \sum_{\mathbf{p}_l^j \in \mathbf{x}_j} \mathbf{w}_l^j\right). \quad (1)$$

Note that we denote kernels over events with capital K and kernels over their component posts using small k.

A very popular kernel choice for text is the linear kernel, which takes the form $k_{\text{text}}(\mathbf{w}_1, \mathbf{w}_2) = \sigma_1^2 \mathbf{w}_1^\top \mathbf{w}_2$ where σ_1^2 is a learned scaling hyper-parameter. Combined with equation (1), the kernel simplifies to

$$K_{\Sigma\text{text}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{\mathbf{p}_k^i \in \mathbf{x}_i} \sum_{\mathbf{p}_l^j \in \mathbf{x}_j} k_{\text{text}}\left(\mathbf{w}_k^i, \mathbf{w}_l^j\right). \quad (2)$$

The kernel in equation (2) is expressed as a double summation of kernels between each pair of posts. It can be also viewed as a convolution kernel between the events, where the decomposition is made into substructures being posts, and combined via simple summation. In this paper we use a linear kernel for the text, although note that we could easily replace k with another kernel for comparing the text, e.g. a string kernel or an RBF kernel, in which case this ceases to be equivalent to a kernel over the concatenated text.

The first non-trivial convolution kernel we consider is the linear kernel normalized by the event sizes,

$$K_{\text{text}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{|\mathbf{x}_i||\mathbf{x}_j|} \sum_{\mathbf{p}_k^i \in \mathbf{x}_i} \sum_{\mathbf{p}_l^j \in \mathbf{x}_j} k_{\text{text}}\left(\mathbf{w}_k^i, \mathbf{w}_l^j\right). \quad (3)$$

We normalize the kernel values so that time series of varying length can be reliably compared without any bias towards longer structures.

A shortcoming of K_{text} is that it operates on text only and ignores time. Therefore, we introduce a convolution kernel using time,

$$K_{\text{time}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{|\mathbf{x}_i||\mathbf{x}_j|} \sum_{\mathbf{p}_k^i \in \mathbf{x}_i} \sum_{\mathbf{p}_l^j \in \mathbf{x}_j} k_{\text{time}}\left(t_k^i, t_l^j\right). \quad (4)$$

This kernel should be useful for comparison of raw time series without additional metadata. For comparing time values we use an RBF kernel, $k_{\text{time}}(t_i, t_j) = \sigma_2^2 \exp\left(-\frac{(t_i - t_j)^2}{l}\right)$, where σ_2^2 is a hyper-parameter controlling the output scale, while $l > 0$ is the length scale, determining the rate at which the kernel diminishes with distance.

We also consider a summation of time and text kernels,

$$K_{\text{text+time}}(\mathbf{x}_i, \mathbf{x}_j) = K_{\text{text}}(\mathbf{x}_i, \mathbf{x}_j) + K_{\text{time}}(\mathbf{x}_i, \mathbf{x}_j). \quad (5)$$

Note that kernel text+time can be viewed as an instantiation of the kernel fusion scheme (Wu, Wu, and Chang 2005) where different views of time series are combined in a weighted average, where weights are kernel variances.

The final convolution kernel we consider compares pairs of posts via a product of kernels over text and time,

$$K_{\text{text}\otimes\text{time}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{|\mathbf{x}_i||\mathbf{x}_j|} \sum_{\mathbf{p}_k^i \in \mathbf{x}_i} \sum_{\mathbf{p}_l^j \in \mathbf{x}_j} k_{\text{text}}\left(\mathbf{w}_k^i, \mathbf{w}_l^j\right) k_{\text{time}}\left(t_k^i, t_l^j\right) \quad (6)$$

This way, the kernel captures not only the textual similarities between posts coming from the two time series, but also how close they are in time. It effectively weights influence coming from each pair of posts by how far apart in time they were created. In this way it can focus on differences in the usage of text across time.

In the *Supplementary Materials*[2] we provide a proof, that those kernels correspond to R-convolutions, which themselves are valid kernels (Haussler 1999).

[2]The *Supplementary Materials* can be accessed at http://people.eng.unimelb.edu.au/tcohn/papers/aaai2016.pdf.

Gaussian Processes

In this section we describe the probabilistic framework of Gaussian Processes (GPs), which we use to demonstrate the usefulness of our kernels. GPs support application to a range of problems, e.g. Poisson regression, allow for learning the hyperparameter values (σ_1, σ_2, l) without expensive cross-validation, and also explicitly model predictive uncertainty.

The Framework

GPs are a Bayesian non-parametric framework that has been shown to work well for a range of problems, often beating other state-of-the-art methods (Cohn and Specia 2013; Beck, Cohn, and Specia 2014). They are widely used for regression, however can also be used for other modeling problems including classification and Poisson regression. The central concept of a GP is a latent function f over inputs $\mathbf{x}$: $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, where m is the mean function, assumed to be 0 and k is the kernel, specifying the degree to which the outputs covary as a function of the inputs.

Let X denote the training set of inputs, $\mathbf{y}$ be the corresponding set of training outputs, $\mathbf{x}_*$ be the test input and $\mathbf{f}_*$ be the corresponding latent function value. The GP posterior is obtained using the prior and the likelihood $p(\mathbf{y}|\mathbf{f})$,

$$p(\mathbf{f}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f})}{p(\mathbf{y}|X)}\,, \tag{7}$$

where in case of regression $p(\mathbf{y}|\mathbf{f})$ is a Gaussian distribution, and it can be used to compute the predictive distribution of latent function values at test data points as

$$p(f^*|X, \mathbf{y}, \mathbf{x}_*) = \int p(f^*|X, \mathbf{x}_*, \mathbf{f})p(\mathbf{f}|\mathbf{y})d\mathbf{f}\,. \tag{8}$$

Non Gaussian Likelihoods

For classification, the latent function is mapped by the squashing function $\Phi(f)$ (in our case probit) into the range $[0, 1]$, such that the result can be interpreted as $p(y = 1|\mathbf{x})$.

For Poisson regression, the likelihood of outputs is given as $p(\mathbf{y}|\mathbf{f}) = \prod_{j=1}^{n} \text{Poisson}\left(y_j|\exp(f_j)\right)$, the Poisson likelihood where the parameter is drawn from the GP, using exponentiation to enforce positivity. The posterior allows us to answer several interesting questions, such as what is the most probable output count for input x_*.

Log-Gaussian Cox Process

Poisson regression allows modeling of a single count variable. However, one may be interested in modeling counts of events in arbitrary subspaces of an input space, e.g. to answer a question like how many tweets arrived in an interval of time in Twitter. A popular method used for point data is a Poisson process, which assumes that the number of events occurring in an interval $[s, e]$ is Poisson distributed with rate $\int_s^e \lambda(t)dt$. A log-Gaussian Cox process (LGCP) (Møller and Syversveen 1998) is an example of a Poisson process, where the intensity function $\lambda(t)$ is modelled such that $\log \lambda(t) \sim \mathcal{GP}(0, k(t, t'))$. We assume an approximation of an integral $\int_s^e \lambda(t)dt \approx (e-s)\lambda(\frac{e+s}{2})$, which allows training the LGCP by applying the Poisson likelihood within the Gaussian Process framework.

Inference

In case of non Gaussian likelihoods, the posterior is intractable and approximation techniques are required. There exist various methods to deal with calculating the posterior. For classification we use Expectation Propagation (EP; (Minka and Lafferty 2002)). In EP, the posterior is approximated by a fully factorised distribution, where each component is assumed to be an unnormalized Gaussian. The parameters of the approximation are iteratively refined to minimise Kullback-Leibler divergence between the true posterior and the approximation. For Poisson regression and LGCP we use Laplace approximation, which approximates the posterior by a Gaussian distribution based on the first two moments of the posterior. Both EP and Laplace approximations yield a Gaussian distribution, making the computations of the posterior over $\mathbf{f}$ tractable.

Hyperparameter Learning

A standard approach to learning hyperparameters in the Gaussian Processes framework is by finding the Type II Maximum Likelihood Estimate on the training set via coordinate descent. This requires finding gradients of the kernel with respect to the hyperparameters $\frac{\mathrm{d}K(\mathbf{x}_i, \mathbf{x}_j)}{\mathrm{d}\theta}$, where $\theta = (\sigma_1^2, \sigma_2^2, l)$ or a subset thereof, depending on the choice of kernel. Each time series kernel is a double summation over pairs of posts, which makes the gradients straightforward to compute.

Experiments on Synthetic Data

We introduce a synthetic experimental setting where we generate events with different temporal, textual and joint temporal-textual characteristics. Our aim is to generate data which is difficult to model using only time or text, but to model the data well requires incorporation of information about text occurrences over time. Below we introduce the generative process we use to simulate the inputs for events $\mathbf{x}_1, \ldots, \mathbf{x}_n$ (in the *Supplementary Materials* we provide more detailed information).

Each event $\mathbf{x}_i$ is generated as a collection of posts, where each post timestamp is drawn iid from a Beta distribution, $t_k^i \sim \text{Beta}(\alpha_i, \beta_i)$, and the text component is generated by drawing each word index from a Geometric distribution, $w_{kl}^i \sim \text{Geom}\left(c_1^i + (c_2^i - c_1^i)t_k^i\right)$, from which a bag-of-words vector representation is computed for each document $\mathbf{w}_k^i$. The hyperparameters $\alpha_i, \beta_i, c_1^i, c_2^i$ differ across events $\mathbf{x}_i$, albeit only slightly, which is achieved by drawing their values from a global prior. The difficulty of discriminative modeling from the events comes from small differences between the generative processes governing them.

We couple the generated events with response variables in three different ways.

Regression: the response variable for event $\mathbf{x}_i$ is set to $y_i = c_1^i - c_2^i$. We use a data set of 100 events with 20 posts each, and evaluate using mean squared error (MSE).

Classification: each event is assigned a binary label $y_i \in \{-, +\}$, and the text generating hyperparameters are tied for events with the same label, i.e., $c_a^i = c_a^\alpha$ where $y_i = \alpha$,

	synthetic data			Ottawa+Ferguson
	classification (ACC↑)	regression (MSE↓)	Poisson regression (LL↑)	classification (ACC↑)
mean	-	0.16 ± 0.08	-39.15 ± 8.18	-
majority	0.44 ± 0.04	-	-	0.701 ± 0.021
text	0.62 ± 0.16	0.15 ± 0.08	-39.46 ± 9.42	0.678 ± 0.025
time	0.44 ± 0.04	0.15 ± 0.07	-39.40 ± 8.21	0.740 ± 0.015
text+time	0.57 ± 0.12	0.14 ± 0.07	-37.01 ± 8.66	0.738 ± 0.011
text ∘ time	**0.80** ± 0.16	**0.05** ± 0.03	**-31.00** ± 5.63	**0.748** ± 0.009

Table 1: Results from the synthetic and rumour data experiments for a range of methods (mean $\pm$ std dev), showing classification accuracy, mean squared error for regression and Poisson log likelihood. In case of synthetic experiments, we ran 55 experiments of each setting, taking 80% of events for training and 20% for testing. In case of the rumour classification experiment, the results are averaged using five-fold cross validation.

for $\alpha \in \{-,+\}, a \in \{1,2\}$. For each label we generate 1000 posts, split into 50 events, and evaluate using predictive accuracy.

Poisson regression: similar to regression, each event has different word generating parameters, and the response variable is set to $y_i = \max\left(1, 1 + \left\lfloor 10(c_1^i - c_2^i)\right\rfloor\right)$. We generate 100 events of 20 posts each and evaluate using the log likelihood (LL) under a Poisson likelihood with parameter equal to predicted mean.

In the three leftmost columns of Table 1 we report results from the three experiment settings. In all settings, text ∘ time yields excellent results, outperforming the other methods. In contrast, the simpler kernels (text, time, text+time) did not significantly improve over the mean or majority baseline. This is because baseline kernels, and most importantly the fusion kernel text+time, do not capture the differences between how the word distributions change over time, an important characteristic of events in our synthetic data.

In Figure 1 we analyze how the performance of the convolution kernels changes as the parameter q controlling the difference of text distributions across events changes (q is the variance of noise perturbing the parameters c_1^i, c_2^i from the common mean; see *Supplementary Materials* for more information about this parameter). Notice that as text marginally differs more across events – due to high variance in appropriate component of generative process – the kernels using text or time only become more and more effective. We observed similar phenomenon when varying the temporal noise term (r), and comparing with the performance of other baseline kernels, i.e., time and text ∘ time. Therefore, the text ∘ time kernel is particularly useful when the differences across events can be found in the temporal variations of text, and are not particularly strong in text or time alone.

Experiments on Social Media Data

We now consider evaluation of our approach on two social media datasets. The two tasks we consider are related to rumour popularity modeling.

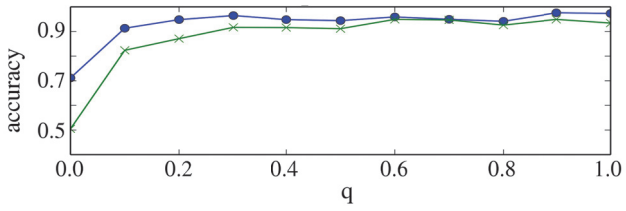

Figure 1: Comparison of accuracy between convolution kernels, showing performance of kernels text ∘ time (blue circles) and text (green crosses) in the synthetic experiments. The plot shows the accuracy as the parameter q controlling the differences across text distributions varies, thus changing the amount of signal in text.

Data

We conduct experiments on two social media datasets. The first dataset consists of 114 popular rumours collected in August 2014 during the Ferguson unrest and is composed of 4098 tweets. This dataset has been used by Lukasik, Cohn, and Bontcheva (2015) for rumour dynamics modeling. We will show how the introduced convolution kernels outperform the state of the art approaches from their work.

We also consider a second dataset consisting of tweets collected October 2014 during Ottawa shootings and in August 2014 during the Ferguson unrest (Zubiaga et al. 2015). The corpus consists of 288 Ferguson rumours and 470 Ottawa rumours, and is composed of 13,002 tweets.

For both datasets, in order to reduce feature sparsity, we replace words with their Brown cluster ids. We used 1000 clusters acquired on a large Twitter corpus (Owoputi et al. 2013), following Lukasik, Cohn, and Bontcheva (2015), whose method we use for benchmarking our results.

Rumour Modeling with Point Processes

The first task is rumour dynamics modeling, introduced by Lukasik, Cohn, and Bontcheva (2015). Consider a set of rumours $\{\mathbf{x}_i\}$, in which each rumour is represented by a set of posts $\mathbf{x}_i = \{\mathbf{p}_k^i\}$. The problem is as follows: given the

	MSE	LL
HPP	7.14±10.1	-23.5±10.1
GP regression	4.58±11.0	-34.6±8.78
Interpolate	4.90±13.1	-
0	2.76±7.81	-
LGCP	3.44±9.99	-15.8±11.6
LGCP ICM	2.46±7.82	-14.8±11.2
LGCP TXT	2.32±7.06	-14.7±9.12
LGCP ICM+TXT	2.31±7.80	-14.6±10.8
LGCP text ∘ time	**2.22**±7.12	**-14.2**±8.7

Table 2: MSE between the true counts and the predicted counts (lower is better) and predictive log likelihood of the true counts from probabilistic models (higher is better) for test intervals over the 114 Ferguson rumours, showing mean ± std. dev. All except the final result are taken from (Lukasik, Cohn, and Bontcheva 2015).

first hour of posts from a rumour $\mathbf{x}_i$, predict the counts of tweets in each of the 6-minutes intervals from the second hour. The predictions are evaluated using two metrics: mean squared error (MSE) over each test interval, and the predictive log-likelihood (LL), used for probabilistic approaches only. Evaluation is run in a leave one out manner, where the prediction is made for each rumour, using the first hour of tweets from the rumour as well as (depending on a method) full two hours of tweets for the remaining 113 rumours.

We reproduce several baselines from (Lukasik, Cohn, and Bontcheva 2015): Homogenous Poisson Process (denoted HPP), Gaussian Process regression (GP), Interpolation and a 'predict no tweets' method (0), all of which are single task methods. The state of the art benchmark methods are based on log-Gaussian Cox Processes. The first approach uses an RBF kernel to independently model each target rumour (denoted LGCP). The remaining methods use multi-task learning over the collections of rumours, using an RBF kernel across time intervals combined in a product with a rumour kernel. The rumour kernel uses the rumour text (TXT) and/or explicit learning of a low-rank matrix of rumour correlations (ICM). For more information about these methods, we refer the reader to (Lukasik, Cohn, and Bontcheva 2015).

We report the results on the 114 Ferguson rumours in Table 2, the same dataset as originally used by Lukasik, Cohn, and Bontcheva (2015). The previous state of the art model was LGCP ICM+TXT and is outperformed by LGCP text ∘ time under both evaluation criteria. Notice how for both metrics the predictive variance is equal to or smaller than the best benchmark results. Thus LGCP text ∘ time not only outperforms the previous best method, but also yields more robust predictions.

Rumour Classification

The second problem is about predicting whether a rumour will be popular, framed as classification. This case study is applicable when authorities or journalists wish to track social media, e.g. for identifying rumours that are going to become popular. Here we use the rumour dataset consisting of 470 Ottawa and 288 Ferguson rumours. As with the earlier rumour dataset, we consider a set of rumours and based on the first hour of posts from rumour $\mathbf{x}_i$, the task is to predict if the number of posts in the second hour equals or exceeds a threshold τ. This results in a binary classification problem, in which we predict which rumours are likely to become or remain popular. For most events we observe a rapid decrease in the counts of posts over time: the mean of number of posts in the first hour is 11.08, and only 1.91 in the second hour. For this reason we set the threshold to $\tau = 2$, which results in around 30% of events being labelled as positive instances.

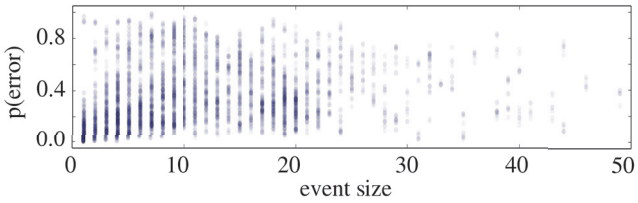

Figure 2: Error analysis versus event lengths (counts of posts in their first hour) for the rumour classification experiment. Showing the probability of error for all events of that length using the GP with the text ∘ time kernel.

We model the data using supervised GP classification, and evaluate the predictive accuracy with 5-fold cross validation, reporting our results in the rightmost column of Table 1. The best accuracy is achieved for the kernel text ∘ time, achieving the score of 0.748. Although this is a small improvement over the mean prediction from the time kernel, notice that the results were much more robust than the other methods (text ∘ time kernel has by far the lowest standard deviation). Overall this is a very difficult modeling problem, although our methods were able to improve over the majority baseline with the best method providing a relative error reduction of 16%. We also notice, that kernel text+time performs worse than text ∘ time, reinforcing our statements from synthetic experiments section that it is a less powerful method in capturing the dynamics of multivariate time series.

Next, we inspect how errors vary across event lengths for the text ∘ time kernel. In Figure 2 we plot points corresponding to events of appropriate size and probability of being assigned a wrong class according to the GP output. Note that there are consistently more events correctly versus incorrectly classified across all event sizes. Even though there is a group of wrongly predicted events of small sizes (upper-left part), there are many more small events that are correctly classified (lower-left part). Surprisingly even short events are overall predicted well, which is a big achievement due to the fact that little data is available for them. On close inspection we found that the text ∘ time kernel shows the greatest improvements over time kernel for shorter events, where presumably text brings much needed information.

Conclusions

In this work we introduced convolution kernels for discriminative modeling from continuous time series of text. We

evaluated the kernels on synthetic data, where we demonstrated intuitive examples where one can expect the kernels to perform well. We also evaluated the kernels on two rumour popularity modeling problems, namely rumour modeling via point processes and rumour popularity classification, and showed how time series convolution kernels outperform the strong baselines. Our proposed convolution kernel can be used in many other potential applications involving discriminative modelling over textual time series, such as activity prediction of social media users or event detection in Twitter.

Acknowledgments

We would like to thank Zsolt Bitvai, Srijith P.K., Daniel Beck and Tomasz Kusmierczyk for helpful discussions and the anonymous reviewers for useful comments. The work was partially supported by the European Union under grant agreement No. 611233 PHEME, and the Australian Research Council Future Fellowship scheme (project number FT130101105). The work was implemented using the GPy toolkit (The GPy authors 2015).

References

Ando, S., and Suzuki, E. 2014. Discriminative learning on exemplary patterns of sequential numerical data. In *ICDM*, 1–10.

Beck, D.; Cohn, T.; and Specia, L. 2014. Joint emotion analysis via multi-task gaussian processes. In *Proc. of EMNLP*, 1798–1803.

Becker, H.; Naaman, M.; and Gravano, L. 2011. Beyond trending topics: Real-world event identification on Twitter. In *ICWSM*.

Bunke, H., and Allermann, G. 1983. Inexact graph matching for structural pattern recognition. *Pattern Recogn. Lett.* 1(4):245–253.

Cohn, T., and Specia, L. 2013. Modelling annotator bias with multi-task gaussian processes: An application to machine translation quality estimation. In *Proc. of the 51st ACL, vol. 1*, 32–42.

Collins, M., and Duffy, N. 2001. Convolution kernels for natural language. In *Proc. of NIPS*, 625–632.

Haussler, D. 1999. Convolution kernels on discrete structures. Technical Report UCS-CRL-99-10, University of California at Santa Cruz.

Jaakkola, T. S., and Haussler, D. 1999. Exploiting generative models in discriminative classifiers. In *Proc. of NIPS*, 487–493.

Jebara, T.; Kondor, R.; and Howard, A. 2004. Probability product kernels. *J. Mach. Learn. Res.* 5:819–844.

Kadous, M. W., and Sammut, C. 2005. Classification of multivariate time series and structured data using constructive induction. *Mach. Learn.* 58(2-3):179–216.

Keogh, E. J., and Pazzani, M. J. 2000. Scaling up dynamic time warping for datamining applications. In *Proc. of 6th KDD*, 285–289.

Liu, X.; Zhou, Y.; and Zheng, R. 2007. Sentence similarity based on dynamic time warping. In *ICSC*, 250–256.

Lodhi, H.; Saunders, C.; Shawe-Taylor, J.; Cristianini, N.; and Watkins, C. 2002. Text classification using string kernels. *J. Mach. Learn. Res.* 2:419–444.

Lukasik, M.; Cohn, T.; and Bontcheva, K. 2015. Point process modelling of rumour dynamics in social media. In *Proc. of the 53rd ACL, vol. 2*, 518–523.

Matuschek, M.; Schlüter, T.; and Conrad, S. 2008. Measuring text similarity with dynamic time warping. In *Proc. of the 2008 IDEAS*, 263–267.

Minka, T., and Lafferty, J. 2002. Expectation-propagation for the generative aspect model. In *Proc. of the 18th UAI*, 352–359.

Møller, J., and Syversveen, A. R. 1998. Log Gaussian Cox processes. *Scandinavian Journal of Statistics* 451–482.

Owoputi, O.; Dyer, C.; Gimpel, K.; Schneider, N.; and Smith, N. A. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proc. of NAACL*, 380–390.

Rabiner, L. R. 1989. A tutorial on hidden markov models and selected applications in speech recognition. In *Proc. of the IEEE*, 257–286.

Rasmussen, C. E., and Williams, C. K. I. 2005. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Shumway, R. H., and Stoffer, D. S. 2006. *Time Series Analysis and Its Applications: With R Examples (Springer Texts in Statistics)*. Springer, 2nd edition.

The GPy authors. 2015. GPy: A Gaussian process framework in Python. http://github.com/SheffieldML/GPy.

Tsuda, K.; Kawanabe, M.; Rtsch, G.; Sonnenburg, S.; and Mller, K.-R. 2001. A new discriminative kernel from probabilistic models. In *Proc. of NIPS*, 977–984.

Vishwanathan, S. V. N.; Schraudolph, N. N.; Kondor, R.; and Borgwardt, K. M. 2010. Graph kernels. *J. Mach. Learn. Res.* 11:1201–1242.

Wu, Y.; Wu, G.; and Chang, E. Y. 2005. Multi-view sequence-data representation and non-metric distance-function learning.

Xing, Z.; Pei, J.; Yu, P. S.; and Wang, K. 2011. Extracting interpretable features for early classification on time series. In *SDM*, 247–258.

Xing, Z.; Pei, J.; and Keogh, E. 2010. A brief survey on sequence classification. *SIGKDD Explor. Newsl.* 12(1):40–48.

Ye, L., and Keogh, E. 2011. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. *Data Mining and Knowledge Discovery* 22(1-2):149–182.

Zubiaga, A.; Liakata, M.; Procter, R.; Bontcheva, K.; and Tolmie, P. 2015. Towards detecting rumours in social media. In *AAAI Workshop on AI for Cities*, 380–390.